Comprehensive Bidding Strategies with Genetic Programming/Finite State Automata

Charles W. Richter, Jr. *
IEEE Student Member

Gerald B. Sheblé *
IEEE Fellow

Dan Ashlock **

* Electrical and Computer Engineering Department
** Mathematics and Complex Adaptive Systems
Iowa State University
Ames, Iowa 50011

Abstract: This research is an extension of the authors' previous work in double auctions aimed at developing bidding strategies for electric utilities which trade electricity competitively. The improvements detailed in this paper come from using data structures which combine genetic programming and finite state automata termed *GP-Automata.* The strategies developed by the method described here are adaptive—reacting to inputs—whereas the previously developed strategies were only suitable in the particular scenario for which they had been designed. The strategies encoded in the GP-Automata are tested in an auction simulator. The simulator pits them against other distribution companies (distcos) and generation companies (gencos), buying and selling power via double auctions implemented in regional commodity exchanges. The GP-Automata are evolved with a genetic algorithm so that they possess certain characteristics. In addition to designing successful bidding strategies (whose usage would result in higher profits) the resulting strategies can also be designed to imitate certain types of trading behaviors. The resulting strategies can be implemented directly in on-line trading, or can be used as realistic competitors in an off-line trading simulator.

Keywords: Competitive auction markets, genetic algorithms, bidding strategies, deregulation, energy broker, power systems, genetic programming, GP-Automata.

I. INTRODUCTION

Regulations governing the electric utility industry in the United States are being changed to promote competition. By increasing competition through deregulation of the transmission network, the Federal Energy Regulatory Commission (FERC) hopes to see increased power system efficiencies and economic benefits for electric consumers.

For decades, electric consumers in the US had only their local vertically integrated utility as a source of electricity. With the passage of the EPAct in 1992, entities that did not own transmission-lines were granted the right to use the transmission system. This was termed *open access* and the US electric utilities began to see limited competition in power production. Countries outside of North America have recently changed their regulation to allow a more competitive electric marketplace. The FERC, in its recent Notices of Proposed Regulation (NOPRs), has announced its intent to expand competition in the US electric marketplace. Attitudes toward re-regulation vary from region to region. California has recently committed itself to adopting a competitive structure and will soon be operating in a manner similar to that described later in this paper.

The research presented in this paper assumes an electric marketplace similar to commodities exchanges like the Chicago Mercantile Exchange, Chicago Board of Trade, and New York Mercantile Exchange (NYMEX) where commodities (other than electricity) have been traded for many years. The fact that in 1996, NYMEX actually added electricity futures to their offerings supports the authors' predictions [12,17,19,20] regarding the coming competitive environment.

In our research, trading agents use a genetic algorithm (GA) to evolve appropriate bidding strategies for the current market conditions. While one could have a complex objective function, the objective of the strategies developed here is strictly profit maximization. These strategies are coded in the form of finite automata coupled with genetic programming (GP-Automata) [2,3]. An optimal bidding strategy must be adaptive, able to properly react as the trading behavior of its competitors changes. The strategies developed in [17] were rigid, akin to fixed steps of a dance, while the strategies we investigate here are adaptive, reacting to changing inputs. Coding information in the form of GP-Automata, which evolve in a GA, allows complex adaptive strategies to develop. The results have been written up specifically for the electricity marketplace, but are directly applicable for other markets.

Part II of this paper surveys recently published work in this area. It discusses research with evolving economic agents, genetic programming applied to auctions, developing successful bidding strategies, deregulation, and the auction environments. Part III describes the methods investigated, including genetic algorithms and GP-Automata. Part IV describes the experiment designed to test the strategies and presents the results of simulation done during this research. Finally, part V presents the conclusions made as a result of the research and lists some possible directions in which this research can be extended.

II. REVIEW OF RECENT WORK

Some work has been done in developing bidding strategies for other electric systems. Finlay [6] analyzed bidding strategies for the restructured Power Pool of England and Wales system, and showed mathematically that there exists an optimal bidding strategy for its bidders. Finlay's work differs in that his objective was not to maximize the profit of the individual generation companies, and the system itself is different from those proposed in the USA.

Post [15] described and compared the various types of basic auctions including the multiple auction, which allows more that one bidder to be awarded a bid, and the double auction where several buyers and sellers submit bids and offers for one unit of a good. When a buyer accepts a seller's offer, or a seller accepts a buyer's bid, a binding contract is made. Post states that "few theoretical results of double auctions exist since modeling strategic behavior on both sides of the market is difficult."

Work by Fahd and Sheblé [5] demonstrated an auction mechanism. Sheblé [20] described the different types of commodity markets and their operation and outlined how each could be applied in the evolved electric energy marketplace. Under the framework described by Sheblé [19], companies presently having both generation and distribution facilities would,

PE-030-PWRS-0-10-1998 A paper recommended and approved by the IEEE Power System Analysis, Computing and Economics Committee of the IEEE Power Engineering Society for publication in the IEEE Transactions on Power Systems. Manuscript submitted January 23, 1998; made available for printing November 10, 1998.

at a minimum, be divided into separate profit and loss centers. Power is generated by generation companies (gencos), transported via transmission companies (transcos), and all power is sold to distribution companies (distcos). Recently the authors proposed that NERC would set the reliability and security standards [23], and predicted that we'll see energy services companies (ENSERVCOs), companies providing ancillary services (ANSILCOs), and energy mercantile associations (EMAs) emerging in this new framework. See Fig. 1 which was presented in [23].

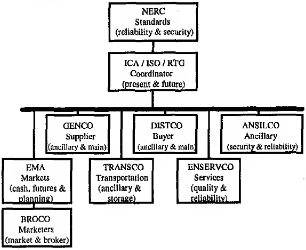

Fig. 1. Brokerage system model.

The framework described by Sheblé [19] allows for a cash market, a futures market and a planning market. The cash market is for trading power for each 30 minute period in the next 30 days. The futures market allows electricity trading from 1 to 18 months into the future. Futures contracts are non-firm for a specific month. This futures market provides a means for electricity traders to manage their risk. The other market is a planning market that can be used to develop capital to build new plants and would allow trading more than 18 months into the future. Sheblé and McCalley [21] outline how cash, future, planning and swap markets can handle real-time control of the system (e.g., automatic generation control) and risk management.

Work by Kumar and Sheblé [11] brought the above ideas together and demonstrated a power system auction game designed to be a training tool. That game used the double auction mechanism in combination with classical optimization techniques. Buyers and sellers interact through a central coordinator, an Independent Contract Administrator (ICA), who matches the bids subject to all operational constraints. The central coordinator is responsible for ensuring that the energy transactions resulting from the matched bids do not overload or render the electrical transmission system insecure. Gencos and distcos coordinate only via the prices transmitted to a central auctioneer. The ICA may submit information to the independent system operator (ISO) or to individual system operators for implementation. The key element is that the ICA is responsible for maintaining the security and reliability of the system. The ICA monitors and responds to the power system limits and transmission capacities. Gencos and distcos are required to cooperate with the ICA in maintaining system reliability. Supplying crucial generator parameters to the ICA during the bidding process is part of this cooperation.

Developing bidding strategies with evolving trading agents for the deregulated electrical utility industry is a new field of research. Apart from the electrical utility industry, interest has grown in recent years for using evolving, or adaptive, agents to simulate trading behavior. Research with adaptive agents has proved to be a useful means of exploring trading markets outside of the electrical industry.

LeBaron [13] uses evolving agents to learn to play financial markets. Tesfatsion [25] describes research in which trading agents decide who to trade with based on an expected payoff. Ashlock, in reference [2], uses genetic programming combined with a finite state automata to play a classic academic game involving bidding behavior and strategies.

Andrews and Prager [1] used a game based on a double auction to verify that genetic search is useful. They show that GP-based agents actually do learn and they compare the performance of the GP-based strategies to those developed using simulated annealing. In addition, they rediscover that at the beginning of the genetic algorithm it is possible to use a less rigorous fitness test than needed in later generations. While their findings may be useful to the genetic algorithm community, their experiments don't realistically model the auction scenario and leave room for further improvements in strategy-building.

III. METHODS AND TECHNIQUES

This section outlines the methods that were used to simulate the marketplace; it introduces the basics of genetic algorithms, genetic programs and GP-Automata; and goes through the process of developing a bid for an auction from a given GP-Automaton.

A. *The Marketplace*

As in [17], the authors again assume the existence of regional commodity exchanges in which buyers and sellers participate in a double auction. This framework has been adopted from Sheblé [19], which is an extension to that being proposed for implementation in California. For the results presented in this paper, transcos are considered to be exogenous to the market, distcos and gencos are allowed to interact in an environment as described in the previous section. Although our framework covers the futures and options markets, we are covering only one market at a time and the results are written up specifically for the cash market. The techniques used to evolve bidding strategies here are general enough to be used in cash, futures or planning markets.

In the double auction used for this research, the bids and offers are sorted into descending and ascending order respectively, similar to the Florida Coordination Group approach as described by Wood and Wollenberg [27]. If the buyer's bid is higher than the seller's offer to be matched, then this is a potentially valid match. The ICA must determine whether the transaction would compromise system security and whether sufficient transmission capacity exists. . If the ICA approves, each potentially valid offer and bid helps to determine the final price, termed the *equilibrium price*. The midpoint price of each pair (weighted by the number of MWs) is used to determine one overall equilibrium price. This is slightly different from the power pool split savings approach that many regions have been using for years in that each of the valid contracts has the same price. An example is given in Table 1. Other pricing mechanisms or market frameworks could be used, but have not yet been investigated for use with the techniques described here.

Table 1. Example of auction bid matching.

Buy bids ($/MW)	Sell offers ($/MW)	Contract?	Midpoint of bid and offer	Equilibrium price ($/MW)
12.50	8.50	Yes	10.50	10.63
12.00	9.00	Yes	10.50	10.63
11.80	10.00	Yes	10.90	10.63
10.00	10.50	No	NA	NA
9.50	11.00	No	NA	NA

In the example shown in the table, there are three bids that are higher than the corresponding offers. If there is not a sufficient number of valid matches, then *price discovery* has not occurred. The auctioneer reports the results of the auction to the market participants. When all bids and offers are collected and insufficient valid bids and offers are found to exist, the auction has gone through one cycle. The auctioneer then reports that price discovery did not occur, and will ask for bids and offers again. The buyers and sellers adjust their bids and offers and another cycle of the auction is played. The cycles continue until price discovery occurs, or until the auctioneer decides to match whatever valid matches exist and continue to the next round or hour of bidding.

After price discovery, the auctioneer asks if another round of bidding is requested. If the market participants have more power to sell or buy, they request another round. Allowing multiple rounds of bidding each time period allows the participants the opportunity to use the latest pricing information in forming their present bid. (One could have a single round with a single bid at each time period, and consider multiple time periods with very little change to the model. This would be similar to theoretical auction research which requires some unrealistic assumptions.) This process is continued until no more requests are received or until the auctioneer decides that enough rounds have taken place. See Fig. 2 for a block diagram of the auction process.

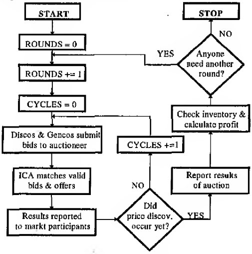

Fig. 2. The auction process.

B. The Basics of Genetic Algorithms

Derived from the biological model of evolution, genetic algorithms (GAs) operate on the Darwinian principle of natural selection [7]. A population of data structures appropriate for the optimization problem are "randomly" initialized. Each of these candidate solutions is termed an individual or a creature. Each creature is assigned a fitness, which is simply a heuristic measure of its quality. Then during the evolutionary process, those creatures that have a higher fitness are favored and allowed to procreate.

During each generation of the evolutionary process, creatures are randomly selected for reproduction with some bias toward higher fitness. After parents are selected for reproduction, they produce children via the processes of *crossover* and *mutation*. The creatures formed during reproduction explore different areas of the solution space than did the parents. These new creatures replace lesser fit creatures of the existing population. The basic algorithm can be written as follows:

1. Randomly initialize a population and set the generation counter to zero.
2. Until done or out of time, do the following:
   - Calculate the fitness of each member of the population.
   - Select parents using some fitness bias.
   - Crossover the parents to create candidate offspring.
   - Mutate these new offspring.
   - Replace the lesser fit members with the offspring.
   - Increment the generation counter and go to step 2.

The parents are required to be in pairs for reproduction, and the result is two children. Children are created by copying the contents of parent 1 into child 1 and of parent 2 into child 2 until a randomly selected crossover location is reached. At this point, bits are copied from parent 1 into child 2, and from parent 2 into child 1.

Following the crossover process, the children are mutated. Mutation introduces new genetic material into the gene at some low rate. If the gene to be mutated in the child is represented by a binary string, mutation involves flipping the bit (0 goes to 1, 1 goes to 0) at each location in the string with some probability. If the gene is represented by an integer, mutation might involve adding an integer that will result in a different valid integer occupying that gene location (loci).

B. The Basics of Genetic Programming

The process of genetic programming has been called automatic programming and is a sub-class of the genetic algorithm field. Genetic programming is a fairly new discipline and is attributed to John Koza [9]. Typically shown in either parse tree (see Fig 3.), or S-expression form (e.g.: split(ite(sub(hbb, cost), 20, asb), 10)). Genetic programs (GPs) are evolvable programs. Each parse tree contains some number of nodes and branches. The branches connect the various nodes which can be either an *operational node* which has arguments and performs some operation involving those arguments, or *a terminal node*.

The designer specifies the set of valid operators and terminals suitable to the problem being investigated. For instance, in developing bidding strategies, suitable operators and terminals might be those described in Table 2. In designing GPs for the GP-Automata, it is desirable to give the trees an opportunity to return numbers in the range of competitive bids.

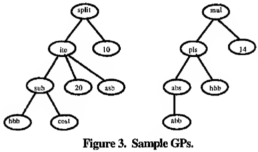

Figure 3. Sample GPs.

Valid GP trees are initialized randomly and then evolved in a standard genetic algorithm (as described in the previous section)

with the following modifications. Crossing over two parents involves randomly selecting a node from each parent and swapping the sub-trees rooted at those nodes. *Mutation* involves randomly selecting a node in the candidate child and throwing away its sub-tree. In its place a new sub-tree is generated randomly. The GP field is new and quite complex at first glance, please see Koza [9,10] for more details.

Table 2. Valid operators and terminals for the GP.

Name	Type	Args	Description
gte	oper	2	Return 1 if 1st arg is >= 2nd arg; otherwise return 0.
ltn	oper	2	Return 1 if 1st arg is <= 2nd arg; otherwise return 0.
ite	oper	3	If 1st arg is even after truncation, return 2nd arg; else, return 3rd arg.
abs	oper	1	Returns absolute value of arg.
mid	oper	2	Returns average of the two args.
mul	oper	2	Returns multiplication of 2 args
pls	oper	2	Returns addition of 2 args
sub	oper	2	Subtracts 2nd arg from 1st arg
max	oper	2	Returns the larger of the two args
min	oper	2	Returns the smaller of the 2 args
cyc	term	0	Returns current cycle number
AOR	term	0	1 if last bid was valid, 0 otherwise
cst	term	0	Returns the cost of gen. for the bid
asb	term	0	Returns the average sell bid
hsb	term	0	Returns the max sell bid
lsb	term	0	Returns the min sell bid
abb	term	0	Returns the average buy bid
hbb	term	0	Returns the max buy bid
lbb	term	0	Returns the min buy bid

C. GP-Automata

GP-Automata combine finite state automata with GP. They were first described as such by Ashlock [2] and were used by Ashlock and Richter [3]. The typical finite state automaton specifies an action and "next state" transition for a given input or inputs. With only one or two binary inputs to work with, it can be fairly simple to develop a finite state diagram to cover the possible input/output relations. When the number of inputs is large the task is much harder. The number of transitions needed to cover all possible combinations of inputs grows exponentially (e.g., 10 inputs each having 5 possible values would require 5.0^{10} transitions). This is where genetic programming comes in. The GP-trees perform bandwidth compression for the GP-Automata by selecting which inputs to consider and performing computations involving these inputs. See Fig. 4 for an example of a GP-Automaton.

Reading the rule encoded by the GP-Automaton in Fig. 4 is fairly simple. We see that this automaton begins by bidding the number in the 'initial action' field. Following the initial action, the 'initial state' tells us which state we would use next (in this case, 2). The GP-Automaton in the figure has four states. Coupled with each of these states is a GP-tree termed a decider. When executed, the decider returns a value between 0-100. Based on that returned value, one of the following two things will happen: (a) if that value is *even* after truncation, the action listed under 'IF EVEN' is taken and we move to the next state listed under 'IF EVEN'; (b) if the returned value is *odd* after truncation, then we use the action and next state listed under 'IF ODD'. The 'action' is the number listed in the action field of the automaton, with two exceptions. The first exception is the 'U' which indicates that the value returned by the decider should be taken directly as the action. The second exception is a '*' which indicates that further computation is necessary and hence the GP-Automata refrains from acting immediately. Instead, it immediately moves to the next state. This gives rise to the possibility of complex (multi-state) computation as well as infinite loops. To prevent infinite loops, after an externally specified maximum number of '*'s, an action is selected at random from actions uniformly distributed over the valid range.

	IF ODD		IF EVEN		
State	Action	Next State	Action	Next State	GP (Decider)
1	14.5	1	U	1	lte (mul (10, abs (hbb))
2	*	1	37	3	ite (max (10, asb), hbb, lbb)
3	12	2	5	1	split (5, abb)
4	U	3	*	2	47
Initial Action		24	Initial State		2

Fig. 4. A four state GP-Automaton.

In this paper we evolve a population of GP-Automata in a GA. The fitness criteria depends on the particular experiment. If one is attempting to evolve strategies that maximize profit, then fitness is profit. After selecting parents as described in the first paragraph of section IV, offspring are produced using crossover and mutation. Crossover for the GP-Automata involves selecting (with a uniform probability) a crossover point ranging from zero to the number of states. We then copy parent1's states from zero to the crossover point to child1 and parent2's states to child2. Following the crossover point, child1 gets parent2's state information and child2 gets parent1's state information (including the associated decider). Before replacing less fit members of the population, each child is subjected to one of four types of mutation. MutationA is standard GA mutation which selects a state or action at random and replaces it with a valid entry. MutationB selects two states within a candidate child and swaps the deciders (intact) associated with these states. MutationC performs the GP crossover as described previously on two states selected randomly from the candidate child. MutationD generates an entirely new decider for a randomly selected state within the candidate child.

D. Using the Strategies in an Auction

In each generation, the performance of each GP-Automata in the population is tested by using the strategy against several competitors. The competitors' bids are initialized randomly each generation of the GA in a Gaussian distribution. This helps to ensure that the resulting GP-Automata strategies will be robust. Based on the competition, some fitness measure (e.g., the profit resulting from the contracts) is assigned to each GP-Automaton.

During the first cycle of bidding in an auction, the strategy defined by the GP-Automata uses the bid specified in the 'Initial Action' cell and goes to the state listed in 'initial state'. For each subsequent cycle of bidding the results of the previous cycle of bidding are available as inputs to the GPs. These inputs are supplied by the terminals listed in Table 2. The GP-trees use the information stored in the terminals as well as numerical terminals in the range 0-100. Bids are taken from the action cell of the automata, except in the cases where the action is listed as a '*' or a 'U', as described previously. The bids are submitted, along with the bids from the competing sellers and buyers, to the auctioneer for evaluation. The bids and offers are matched and a would-be price is reported, completing one cycle of the auction. The cycles continue until price discovery occurs or until some maximum number of cycles (maxcycles) has passed. There is a maxcycles parameter,

which is selected uniformly over a range to prevent the strategies from falling into a local optima in which the strategies work well when the number of cycles is identical over the trials in a given generation.

IV. RESULTS

The following parameters were used for the results presented in this section. The population size (i.e. number of GP-Automata in the population) was set to 24. We used the roulette parent selection method, which probabilistically selects parents in proportion to their fitness. Each generation, we replace the 8 least-fit in the population. During every generation of the GA, each GP-Automaton's fitness was calculated by having it participate as a seller (genco) in an auction 50 times with 5 buyers demanding 15 MWs and 5 sellers offering to supply 10 MWs. Fitness was equal to the profit garnered by using the strategy in those 50 auctions. Each automaton is also attempting to sell 10 MWs. The $5/MW cost associated with generating the 10 MWs is the same for each automaton. Each automaton being tested faces the same 50 sets of bids and offers that the other automata face. The GA was allowed to evolve for 35 generations. Each experiment/case was repeated 20 times to test the robustness of the method, and the results shown in this section are averaged across those twenty runs. Since we are attempting to build generic bidding rules, we test the rules in a variety of scenarios, hence the need for as many independent experiments as time allows.

In case 1, the buyers' bids are all set to a constant $15/MW, and the sellers' offers are set to a constant $5/MW. Fig. 5-a shows the min, max, and average fitness of the automata in each generation of the GA. Because (as described previously) the maximum number of cycles is not always the same, the max fitness will sometimes decrease from one generation to the next. Figure 5-b shows a histogram of the bids placed by our GP-Automata as they participate in the auction. We can see that the GP-Automata are learning to bid close to their cost of generation, which is $5/MW (corresponds to 25 on the histogram). (**Note**: In Figs. 5, 6, & 7, the bids are shown ranging from 0 to 100 along the x-axis. A bid shown in the histograms as 100, corresponds to $20/MW, while 50 corresponds to $10/MW, and so forth.)

In case 2 (shown in Fig. 6), the offers of the competing sellers are distributed about $5/MW and the buyers' bids are distributed around $15/MW as shown in Fig. 6-c. This makes it harder for the GP-Automata, and their maximum fitness is lower than that of case 1. Their bids tend to be at or below cost, which when matched with buy bids as shown in Fig. 6-c. will result in some amount of profit. As in cases 1 and 3, they learn very quickly that they must not be underbid by the other generators, hence we see no successful strategies with bidding above $5.

In case 3 (shown in Fig. 7), we use the same competing bid distribution for the first cycle of each auction, but the bids decrease and the offers increase with each passing cycle making it a bit more competitive. The histogram for case 3 indicates that the bidding of our automata becomes a bit more exploratory. Since the competing gencos' offers are increasing, the GP-Automata also have room to increase their offers as the cycles continue in an attempt to increase their profits.

Notice that the profit obtained by the GP-Automata decreases as the bids and offers of their competitors become more noisy. The profit decreases even more as the bidding becomes more competitive as in case 3. Figs. 5-a, 6-a, and 7-a indicate maximum fitness doesn't change drastically. Since we have 24 automata being initialized uniformly over the entire range, it is highly probably that at least one of these automata will make a bid close to the optimal. The graph of average fitness indicates that the rest of the population quickly learns to bid similarly in subsequent generations.

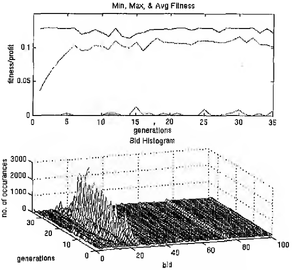

Figure 5 (a, b). Case 1. Competitor buy bids at $15/MW & sell offers at $5/MW.

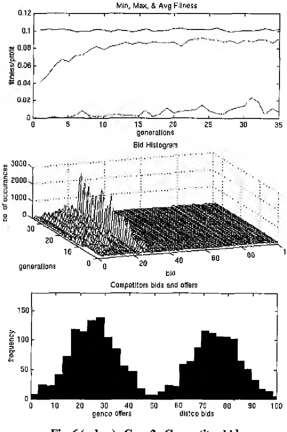

Fig. 6 (a, b, c). Case 2. Competitor bids distributed as shown in (c).

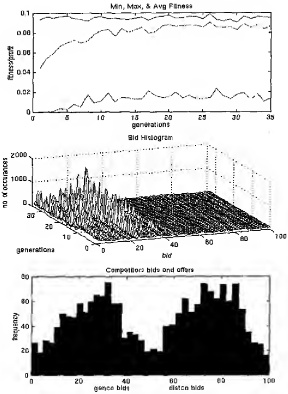

Fig. 7 (a, b, c). Case 3. Competitors' bids/offers moving toward each other with each passing cycle of the auction.

V. CONCLUSIONS & FUTURE RESEARCH

The results demonstrate that GP-Automata learn to bid in a sensible and explicable manner. The GP-Automata lend themselves well to scenarios where there are vast amounts of data available, and identification of crucial data is important. The company models used in the simulations described in this paper were fairly straight forward. Adding more detail (e.g. ATCs, forecasted prices, unit commitment schedules) will increase the volume of information that the bidder needs to consider in making a bid. We plan to test in further experiments that the GP-Automata are able to make use of this additional data to increase the performance of the strategies.

The sensitivity of the solution to the selection of the various parameters is a very important area that needs investigation. Many of the parameters used for the experiment described here were set using engineering judgement and should be verified as being appropriate. The authors are performing research designed to test the sensitivity of many of the parameters. Among these parameters are the parent selection methodology, the population size, and modifications on the auction. A paper describing the effects of reducing the number of states and number of decider nodes has already been published [18].

VI. REFERENCES

[1] M. Andrews and R. Prager, "Genetic Programming for the Acquisition of Double Auction Market Strategies." *Advances in Genetic Programming*. Edited by K. Kinnear, Jr. Cambridge, Massachusetts: The MIT Press, 1994.

[2] D. Ashlock, "GP-automata for Dividing the Dollar." Mathematics Department, Iowa State University, Ames, IA 1995.

[3] D. Ashlock and C. Richter, "The Effect of Splitting Populations on Bidding Strategies." GP-97 Proceedings of the Second Annual Conference, pp. 27-34. San Francisco, CA: Morgan Kaufmann Publishers.

[4] B. Barkovich, and D. Hawk, "Charting a new course in California", *IEEE Spectrum*, Vol. 33, July 1996, p26.

[5] G. Fahd, and G. Sheblé, "Optimal power flow emulation of interchange brokerage systems using linear programming," *IEEE Transactions on Power Systems*, Vol. 7, No. 2, pp. 497-504, 1992.

[6] D. Finlay, *Optimal Bidding Strategies in Competitive Electric Power Pools.* Masters thesis, University of Illinois, Urbana-Champaign, IL, 1995.

[7] D. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.

[8] Y. Ioannides, "Evolution of Trading Structures." Department of Economics, Tufts University, Medford, MA 02155, 1995.

[9] J. Koza. *Genetic Programming.* Cambridge, Massachusetts: The MIT Press, 1992.

[10] J. Koza. *Genetic Programming II.* Cambridge, Massachusetts: The MIT Press, 1992.

[11] J. Kumar, G. Sheblé, "Auction Game in Electric Power Market Place." *Proceedings of the 58th American Power Conference*, 1996, pp. 356-364.

[12] J. Kumar, G. Sheblé, "Framework for Energy Brokerage System with Reserve Margin and Transmission Losses." 1996 IEEE/PES Winter Meeting, 96 WM 190-9 PWRS. New York: IEEE

[13] B. LeBaron, "Experiments in Evolutionary Finance." University of Wisconsin - Madison, Madison, Wisconsin, 1995.

[14] P. Milgrom, *Auctions and Bidding: A Primer.* Journal of Economic Perspectives, Vol. 3, No. 3. Summer 1989, pp. 3-22, 1989.

[15] D. Post, *Electric power interchange transaction analysis and selection.* Master's thesis, Iowa State University, Ames, IA, 1994.

[16] C. Richter, *Developing bidding strategies for electric utilities in a competitive environment.* Master's thesis, Iowa State University, Ames, IA, 1996.

[17] C. Richter and G. Sheblé, "Genetic Algorithm Evolution of Utility Bidding Strategies for the Competitive Marketplace." 1997 IEEE/PES Summer Meeting, PE-752-PWRS-1-05-1997. New York: IEEE

[18] C. Richter, D. Ashlock and G. Sheblé, "The Effects of State Number and Tree Size on Bidding Strategies." Proceedings of the 1998 Conference on Genetic Programming, 1998.

[19] G. Sheblé, "Priced Based Operation in an Auction Market Structure". Paper presented at the 1996 IEEE/PES Winter Meeting. Baltimore, MD, 1996.

[20] G. Sheblé, "Electric energy in a fully evolved marketplace." Presented at the 1994 North American Power Symposium, Kansas State University, KS, 1994.

[21] G. Sheblé, and J. McCalley, "Discrete auction systems for power system management." Presented at the 1994 National Science Foundation Workshop, Pullman, WA, 1994.

[22] G. Sheblé "Simulation of discrete auction systems for power system risk management." Frontiers of Power, Oklahoma, 1994.

[23] G. Sheblé, M. Ilic, B. Wollenberg, and F. Wu. Lecture notes from: *Engineering Strategies for Open Access Transmission Systems,* A Two-Day Short Course presented Dec. 5 and 6, 1996 in San Francisco, CA.

[24] G. Thompson, and S. Thore, *Computational Economics: Economic Modeling with Optimization.* San Francisco, CA: Scientific Press, 1992.

[25] L. Tesfatsion, "A Trade Network Game with Endogenous Partner Selection." *Economic Report Series*, Department of Economics, Iowa State University, Ames, IA, 1995.

[26] A. Vojdani, C. Imparto, N. Saini, B. Wollenberg, and H. Happ, "Transmission Access Issues." Presented at the 1995 IEEE/PES Winter Meeting, 95 WM 121-4 PWRS. New York: IEEE, 1994.

[27] A. Wood, and B. Wollenberg, *Power Generation, Operation, and Control.* New York, NY: John Wiley & Sons, 1984.

VII. BIOGRAPHIES

Charles W. Richter, Jr. (M '89) received his BS-EE from South Dakota State University in 1992, and his MS-EE from ISU in 1996. He has worked for an electric utility and a consulting firm and is presently completing a Ph.D. degree in Electrical Engineering at Iowa State University. Research interests include power system operations, economics, and complex adaptive systems.

Gerald B. Sheblé (M 71, SM 85, F 97) is a Professor of EE at ISU, Ames, IA. Dr. Sheblé received his BS and MS degrees in EE from Purdue University, and his Ph.D. in EE from Virginia Polytechnic. His more than fifteen years of industrial experience include projects with public utilities, research and development, computer vendors and consulting firms. His research interests include power system optimization, power system economics, scheduling and control.

Daniel Ashlock is an Associate Professor of Mathematics at ISU. He earned his BS degrees in math and in computer science from the University of Kansas in 1984. In 1990, Dr. Ashlock received his Ph.D. in math from CalTech. His research and academic interests include combinatorics, graph theory, computer applications to discrete mathematics, and complex adaptive systems.